fltkhs-kerho

Itsearviointi ja jälkiselvitys

Tiivistelmä ohjelmointityöstä

Ohjelmointityö kurssille TIEA306 tehty kerho-ohjelma toteutus käyttäen haskell:ia

Tehtävän kuvaus, tausta ja tavoitteet

Tehtävänä oli luoda kopio vesan tekemästä kerho-ohjelmasta käyttäen kielenä haskell:ia. Valitsin aiheen, koska mielestäni haskell on mielenkiintoinen kieli ja halusin selvittää miten osaisin toteuttaa sillä ihan omin neuvoin interaktiivisen ohjelman. Tavoitteeni oli saada aikaiseksi ohjelma, joka täyttäisi vesan vastaavan ohjelman perusominaisuudet.

Käytännön toteutus

Ohjelman lähdekoodi muodostuu 4 komponentista: fltkhs-kerho.hs (eli main), joka kutsuu fluid tiedostossa määriteltyä käyttöliittymää. Fluid tiedosto Kayttoliittyma.fl, joka määrittelee käyttöliittymän ulkonäön (tämän tiedoston editoinnista/tarkastelusta lisää myöhemmin). Callbacks.hs joka sisältää kaikki eventtien eli callbackkien kutsumat toiminnot, sekä pitää yllä tietorakennetta, jota se muokkaa käyttöliittymästä saatujen inputtien mukaan, sekä hoitaa tiedostoihin tallentamisen ja tiedostojen avaamisen, eli lyhyesti, hoitaa reagoinnin ohjelman käyttäjän toimintaan. Ja lopuksi Tietorakenteet.hs, joka määrittelee tietorakenteet ja sisältää tietorakenteiden muokkaamiseen ja tarkistamiseen käytettäviä funktioita. Kaikki neljä tiedostoa ovat minun itseni kirjoittamia, mutta Callback.hs sisältää harrastusten infotaulukon, joka on otettu fltkhs tekijän demosta pienin muokkauksin (kyseinen on selkeästi eroteltu ja merkitty lähdekoodiin, sekä julkaistu MIT-lisenssin alla, joka käsittääkseni antaa minulle vapaat kädet kyseisen koodin käyttöön).

Oman työn arviointi

Ihan alkuperäinen suunnitelmani olisi ollut että olisin saanut tehtyä hieman nätimmän ja paremman ohjelman

(jouduin karsimaan alkuperäisestä ohjelmasta löytyvän tulostamisen ja harrastusten muokkaamisen. Myös harrastustaulukon päivittyminen heti harrastuksen poiston/lisäämisen jälkeen jäi uupumaan. Tietorakenne kyllä päivittyy heti, mutta jäsen joudutaan valitsemaan uudestaan tai poistamaan/lisäämään toinen harrastus, jotta itse lista päivittyisi, syytä sille miksei redraw komenno toimi en tosin tiedä) Myös Jäsenten harrastuksia listaavan taulukon sort ei toimi kunnolla muussa kun lajin kohdassa. Aloitusvuoden ja h/vko kohtien osalla numeroita verrataan myös teksteinä, joten esimerkiksi 11 tulee ennen 9. Myös ulkonäöllisesti ohjelma ei ole mitään silmäkarkkia, mutta jos fltkhs perustuu n. 20 vuotta vanhaan fltk, en tiedä voinko parempaa odottaakkaan.

Eniten vaikeuksia tuotti lähdemateriaalin puute. FLTKHS dokumentointi on ainakin minusta hieman liiankin vähäsanaista. Funktioista annetaan vaan nimet ja syötteiden tyyppi, eikä mitään esimerkkiä tai selitystä miten ne toimivat. Joissain tapauksissa tämä riittää, mutta itse huomasin että jouduin kokeilemaan tekikö funktio sitä mitä oletin sen nimen perusteella tekevän. Mielestäni tämä ei ole hyvää ajankäyttöä. Myös haskellin tyyppisysteemi tuotti välillä ongelmia, mutta suurimmasta osasta ajasta se oli tielläni ihan syystä, ja myönnän että kyseinen auttoi karsimaan virheitä huomattavasti. Suurin ongelma tyyppien kanssa oli erityisesti String ja Text tyypin kanssa. Puolet kääntämisessä saamista virheilmoituksista tulivat näistä, ja pyrin vielä pysymään vain Text:issä, koska fltkhs tuki sitä. Hyvänä esimerkkinä tästä on se että TEXT paketin tiedostosta luku tahtoo tiedoston sijainnin String:inä eikä Text:inä. Yksi tyyppi teksteille on tarpeeksi. Odotin että fltkhs tulisi olemaan suurin ongelmienlähde, mutta yllätykseni paketti itsessään on aivan mainio väline, jahka alkukankeudesta päästiin eroon.

Olen etsinyt ohjelmastani tapoja joilla sen saisi kaatumaan toistettavasti ja olen pyrkinyt paikkaamaan niitä. Ainoat kaatumiset joista tiedän, johtuvat siitä jos kerhojen tallennustiedostoja käydään manuaalisesti muuttamassa ja yritetään sen jälkeen ladata sitä. Voi olla että harrastuksien poistoon on jäänyt vieläkin jokin kaatumisen aiheuttava virhe, mutta mielestäni kyseiset olen jo metsästänyt läpi, tosin en ole varma, koska suurinosa kyseisestä koodista ei ole minun kirjoittamaa (harjoitustaulukko), ja sen toiminta on osin minulle vieläkin joiltain osin mysteeri.

Työkalut eivät nyt mitenkään erityisesti haitanneet tai auttaneet minua työssä, paitsi fluid. Ajattelin aluksi että käyttöliittymänsuunnitteluohjelma olisi todella kätevä työtä tehdessä, mutta se aika mitä säästin komponenttien asettelussa, jouduin käyttämään moninkertaisesti saadakseni fluidissa laitetut komponentit toimimaan halutusti. Ja koska Käyttöliittymä.fl määrittelee käyttöliittymän, jouduin ahtamaan pienen osan koodista .fl tiedostoon (joitakin komponentteja ei löydy suoraan fluidista, vaan ne joudutaan lisäämään sinne koodaamalla), joka tarkoittaa että kyseisen koodin lukemiseen ja muokkaamiseen joudutaan käyttämään fluidin omaa käyttöliittymää, joka ei ole ihan optimaalinen.

Aikataulu oli ISOIN akilleenkantapääni ylivoimaisesti. Pyysin aihetta joulukuussa, enkä ollut käytännössä saanut mitään aikaiseksi vielä huhtikuussa, ja huhti-toukokuu väli oli myös hyvin hiljaista työn teon osalta. Kesäkuussa sain hyvin tuulta työni purjeisiin, mutta silloinkin kun jäin pahaan paikkaan työni edistyminen tyssäsi taas. Vasta heinäkuussa sain todella työtä edistettyä, ja sen jälkeen työtahtini onkin mielestäni ollut ihan hyvää, aloitukseni oli vaan suoraan sanottuna aivan kamala. Olen koko yliopistoajan ollut todella huono tekemään asioita ajallaan ja jätänkin ne yleensä viime tinkaan. Pitäisi petrata paljon tältä osastolta ja vaan yrittää pakottaa itseni tekemään asioita. Yleensä kun saan edes vähän edettyä työssä niin jatkaminen on helppoa.

Verrattuna kursseilla opetettuun ja kirjallisuuteen mielestäni työn käytännössä tekeminen (ainakin näin

sooloprojektina) on huomattavasti hektisempää mitä on annettu odottaa. Kursseilla on mielestäni annettu ohjelmoinnista sellainen kuva että sinulla on suunnitelma jonka mukaan teet ja ohjelma rakentuu pikkuhiljaa pala palaltaan kokoon. Omassa työskentelyssäni huomasin että olen koodannut itseni umpikujaan ja jouduin purkamaan aijempia rakennelmia ja tekemään tilalle uusia, tai yrittää saada vanhat viritelmät toimimaan jollain uudella tavalla, joka silloin tällöin johti siihen että jouduin myöhemmin tekemään vielä ihmeellisempiä viritelmiä, jotta saisin työni toimimaan.

Työssä käytetyt lähteet

Työn lähteinä on käytetty fltkhs ja fltk dokumentointia, sekä otettu mallia ja apinoitu fltkhs:lletehdyistä demoista. Materiaalia on netissä yllättävän vähän dokumentointien lisäksi, joka olikin mielestäni haastavin tekijä koko työn teossa.

Ylläpitäjän tarvitsemat tiedot

Kääntäminen

Readme sisältää hyvin helposti seurattavan kääntämisohjeen, joka on paremmin jäsennelty. Ohjelman kääntämiseen tarvitaan stack. Stackin asentamisen jälkeen käyttäjän tarvitsee onnistua kääntämään fltkhs-hello-world demo fltkhs (ohjeet löytyvät linkistä). Olen henkilökohtaisesti onnistunut kääntämään ohjelman Windows 7/10 muutamalla eri koneella, ja paketin tekijän mukaan kääntäminen pitäisi olla helpompaa muilla tuetuilla alustoilla, koska toisin kuin Windowsin tapauksessa, käyttäjän ei erikseen tarvitse asentaa esim. `autotool` tai `tar` paketteja. Windows käyttäjien tarvitsee kuitenkin ladata kyseiset paketit stackin mukana tulevalla msys2 shellillä, jonka jälkeen käyttäjät voivat vasta kääntää ohjelman (ohjeet tähän löydät fltkhs linkistä). Huomioitavaa: jotkut antivirukset (kuten avast) merkkaavat jonkun msys2:lla ladattavista paketeista karanteeniin. Tämän tapahduttua stack ei voi enää kääntää ohjelmaa, ja ainoa korjaustapa jonka olen todennut toimivaksi on poistaa stack kokonaan, laittaa antiviruksen pois päältä ja kokeilemalla koko touhua alusta pakettien lataus mukaanlukien. Omassa tapauksessani msys2 kansio jäi kummittelemaan stackin poiston jälkeenkin asennuskohteeseen, eikä suostunut deletoitumaan ennen kun käynnistin koneen uudelleen. Onnistuin myös toistamaan kyseisen kaverini koneella, joten suosittelen ainakin avastin kohdalla laittamaan kyseisen pois päältä kääntämisen ajaksi, jos aiot kääntää tätä ohjelmaa. Suosittelen myös suht vakaata nettiyhteyttä, koska pätkivä netti on toinen asia jonka on onnistunut itsellä pilaamaan stackin toiminnan.

Jos olet onnistunut kääntämään fltkhs-hello-world ohjelman, pitäisi kerho-ohjelman kääntäminen onnistua myös komentorivillä yksinkertaisesti menemällä kohdekansioon ja kääntämällä lähdekoodi ohjelmaksi komennolla `stack install --flag fltkhs:bundled`. Ensimmäisellä kääntökerralla stack tekee paljon asioita .stack-work kansioon, johon ohjelma käännetään. Kun tämän lisää stackin hitaisii latauksiin ensimmäinen kääntäminen tulee kestämään useita minuutteja (9 min kun kerran jaksoin mitata), joten maltti on valttia. Tulevat kääntökerrat tosin ovat huomattavasti nopeampia (n. 20 sekunttia). Lopuksi ohjelman voi käynnistää komennolla `stack exec fltkhs-fluid-kerho` (ohjelman .exe tiedoston käynnistäminen suoraan ei toimi, koska oletuksena vaadittavat linkitetyt tiedostot eivät löydy kansiosta johon ohjelma on käännetty, jos haluat jakaa ohjelmaa ilman lähdekoodia, fltkhs linkissä on siihenkin ohjeet, mutta itse en ole kyseistä kokeillut). Src kansiosta löytyvää .fl tiedostoa, joka kääntämisen yhteydessä käännetään haskelliksi fltkhs avulla, pääsee tutkimaan fltk mukana tulevalla gui editorilla, fluidilla. Fltk paketin kääntäminen onnistuu esim. visual studiolla.

Tiedettyjä oikkuja

Välillä käynnistäessä ohjelma antaa virheilmoituksen `freeHaskellFunctionPtr: not for me, guv!` ja kaatuu. Syytä tähän en tiedä, mutta ohjelman uudelleenkäynnistäminen toimii. Kääntäjä ei huomaa fluid (.fl) tiedostoon tehtyjä muokkauksia automaattisesti, joten jos muutat jotain siellä joudut myös muuttamaan jotain toista tiedostoa, jotta kääntäjä suostuu kääntämään ohjelman uudestaan.

Ohjelman käyttäminen

Ohjelman menubaarista löytyy "Apua" painike, joka antaa tiiviin infopaketin siitä minkälaisia syötteitä tietyt kentät hyväksyvät tallentamaan, jos kyseisellä syötteellä on olemassa jonkinlainen tarkistus olemassa. Paketin mukana tulee myös "malli" kerho, jonka käyttäjä voi ladata kirjoittamalla "Avaa" menunappulasta tulevaan ikkunaan "malli" ja painamalla nappia. Malli kerho antaa esimerkkejä siitä, minkälaisia syötteitä mitkäkin kentät vaativat, niin hyvässä kuin pahassa.

Mahdollinen jakokehitys

Jos joku tahtoo jatkaa kyseistä projektia, seuraava luonnollinen vaihe olisi tietenkin korjata pari ohjelmasta löytyvää vikaa ja implementoida alkuperäisestä kerho-ohjelmasta lötyvät, mutta tästä puuttuvat toiminnot työhön (kuten erillinen harrastusten muokkaus). Myös fltkhs themeihin tutustumalla, voi käyttöliittymästä saada vähän parempaa silmäkarkkia, mutta mielestäni kirjoitushetkellä kyseisiä ei olla vielä kauheasti implementoitu paketin tekijän osalta

Yhteenveto

Tekemällä oppii parhaiten, vaikka se vaatiikin myös eniten työtä. Mielestäni tällaisillä tekemiseen painoitetuilla kursseilla oppii paljon paremmin, vaikka ne ovatkin myös huomattavasti luentokursseja työläämpiä tehdä. Ja verrattuna aijempiin isompii ohjelmointiprojekteihin tämä tuli tehtyä yksin, joka vielä edistää oppimista ohjelmoinnin osalta lisää. Loppuenlopuksi olen ihan tyytyväinen aikaan saamaani työhön, ja vaikka aloittaminen tuntui aivan kamalalta, loppua kohden työn tekeminen alkoikin parhaimmillaan olla ihan mukavaa.